



Algorithms for the Multiclass Network Equilibrium Problem

Thomas Pradeau, Frédéric Meunier

► To cite this version:

Thomas Pradeau, Frédéric Meunier. Algorithms for the Multiclass Network Equilibrium Problem. ROADEF - 15ème congrès annuel de la Société française de recherche opérationnelle et d'aide à la décision, Société française de recherche opérationnelle et d'aide à la décision, Feb 2014, Bordeaux, France. hal-00946410

HAL Id: hal-00946410

<https://hal.science/hal-00946410>

Submitted on 13 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ALGORITHMS FOR THE MULTICLASS NETWORK EQUILIBRIUM PROBLEM

THOMAS PRADEAU AND FRÉDÉRIC MEUNIER

1. THE PROBLEM

Being able to predict the impact of a new infrastructure on the traffic in a transportation network is an old but still important objective for transport planners. In 1952, Wardrop [7] noted that after some while the traffic arranges itself to form an equilibrium. With the terminology of game theory, the equilibrium is a Nash equilibrium for a nonatomic congestion game. In 1956, Beckmann [1] translated these principles as a convex mathematical program, opening the door to the tools from convex optimization. One of the main assumptions used by Beckmann is the fact that all users are equally impacted by congestion: there is only one *class*. In order to improve the prediction of traffic patterns, researchers started in the 70s to study the *multiclass* situation, each class modeling a distinct mode of transportation, such as cars, trucks, or motorbikes. Except in very few special cases, no algorithm is known for this problem.

1.1. Model. We are given a directed graph $D = (V, A)$ modeling the transportation network. The population of *players* is modeled as a bounded real interval endowed with the Lebesgue measure, the *population measure*. Each player wants to go from its origin vertex to its destination vertex at the minimum cost, the cost of an arc a depending on the total flow x_a on it, defined as the measure of the players going through this arc. The set of players is partitioned into a finite number K of measurable subsets – the *classes* – modeling the players with same characteristics: they share a same collection of cost functions $(c_a^k : \mathbb{R}_+ \rightarrow \mathbb{R}_+)_{a \in A}$, a same origin s^k , and a same destination t^k . The set of vertices (resp. arcs) reachable from s^k is denoted V^k (resp. A^k).

A strategy profile is a mapping determining for each player a route from its origin to its destination. A strategy profile is a (pure) Nash equilibrium if no player would benefit by changing its route.

This game enters in the category of *nonatomic congestion games with player-specific cost functions*, see Milchtaich [5]. In particular, under mild conditions, an equilibrium is known to exist. The problem of finding a Nash equilibrium for such a game is called the *Multiclass Network Equilibrium Problem*.

1.2. Contribution. We consider the specific case where the cost functions are affine and strictly increasing:

$$c_a^k(x) = \alpha_a^k x + \beta_a^k, \text{ with } \alpha_a^k > 0, \beta_a^k \geq 0,$$

for each class k and arc a .

- We prove the existence of a polynomial algorithm solving the problem when the graph consists in parallel arcs, for a fixed number of classes. The main idea of the algorithm relies on properties of hyperplane arrangements. This kind of congestion games, also called *parallel-link games*, have been extensively studied (see [3] and references therein).
- We build a pivoting algorithm, adapted from Lemke [4], solving the problem for general graphs. To the best of our knowledge, it is the first algorithm solving this problem. We prove its efficiency through computational experiments. Moreover, this algorithm provides the first constructive proof of the existence of an equilibrium for this problem.

Key words and phrases. Affine cost functions; congestion externalities; constructive proof; Lemke algorithm; nonatomic games; transportation network.

2. A POLYNOMIAL ALGORITHM FOR PARALLEL-LINK GAMES

In this section, we suppose that D has only two vertices $V = \{s, t\}$ and that all arcs are parallel with origin s and destination t . We identify then arcs and routes. For a flow $\mathbf{x} = (x_a^k)_{k \in K, a \in A}$, we define for every class k its *support* as the set of arcs with positive flow: $\text{supp}(\mathbf{x}^k) = \{a \in A, x_a^k > 0\}$.

Lemma 1. *Given a subset of arcs $S^k \subseteq A$ for each class k , it can be decided in polynomial time whether the S^k 's are the supports of an equilibrium flow, and, if the answer is 'yes', an equilibrium flow can be computed in polynomial time.*

Indeed, the equilibrium conditions form then a system of linear inequalities that can be solved polynomially with the interior point algorithm.

Lemma 2. *For a fixed number of classes, we can determine in polynomial time a set $\mathcal{S} = \{(S^k)_{k \in K} : S^k \subseteq A\}$ of polynomial size such that $(\text{supp}(\bar{\mathbf{x}}^k))_{k \in K} \in \mathcal{S}$, where $\bar{\mathbf{x}}$ is an equilibrium flow.*

We sketch the proof of Lemma 2.

For an equilibrium flow $\bar{\mathbf{x}}$, we denote by $\gamma^k(\bar{\mathbf{x}})$ the cost at equilibrium of the class k players. Then, we have by definition

$$(1) \quad \begin{aligned} c_a^k(\bar{x}_a) &\geq \gamma^k(\bar{\mathbf{x}}) && \text{for all } a \in A, \\ c_a^k(\bar{x}_a) &= \gamma^k(\bar{\mathbf{x}}) && \text{for all } a \in \text{supp}(\bar{\mathbf{x}}^k). \end{aligned}$$

A crucial remark is that when an arc a belongs to the support of two different classes $k_1 \neq k_2$, we must have according to Equation (1)

$$\bar{x}_a = (c_a^{k_1})^{-1}(\gamma^{k_1}(\bar{\mathbf{x}})) = (c_a^{k_2})^{-1}(\gamma^{k_2}(\bar{\mathbf{x}})).$$

It means that the vector of equilibrium costs $\boldsymbol{\gamma}(\bar{\mathbf{x}}) \in \mathbb{R}^K$ belongs to the hyperplane of \mathbb{R}^K :

$$H_a^{k_1, k_2} = \{\mathbf{y} \in \mathbb{R}^K, \alpha_a^{k_1}(y^{k_2} - \beta_a^{k_2}) = \alpha_a^{k_2}(y^{k_1} - \beta_a^{k_1})\}.$$

We define then a set \mathcal{H} of hyperplanes of this kind and consider the arrangement of hyperplanes associated. The cells provide a set \mathcal{S} of candidate supports. Lemma 2 follows from the fact that there is a polynomial number $\Theta((K^2|A|)^K)$ of cells in fixed dimension (see for example [2]).

Applying repeatedly the algorithm given by Lemma 1 on each cell leads to the following theorem.

Theorem 1. *For a fixed number of classes, there exists an algorithm solving the Multiclass Network Equilibrium Problem in polynomial time with respect to the number of arcs.*

3. A LEMKE-LIKE ALGORITHM FOR GENERAL NETWORKS

3.1. Formulation as a linear complementarity problem. In the single-class case, the equilibrium flow is an optimal solution of a convex optimization problem, see Beckmann et al. [1]. If the flows $\mathbf{x}^{k'}$ for $k' \neq k$ are fixed, finding the equilibrium flow \mathbf{x}^k for the class k is again a single-class problem. With the help of the Karush-Kuhn-Tucker conditions, we get that the equilibrium flow \mathbf{x} coincides with the solutions of a complementarity problem. When the cost functions are affine, it becomes a linear complementarity problem.

Similarly as for the Lemke algorithm [4], we rewrite the problem as an optimization problem using a vector $\mathbf{e} = (e_a^k)_{k \in K, a \in A^k}$ and an extra variable ω .

Proposition 1. $(\mathbf{x}^k)_{k \in K}$ is an equilibrium flow if and only if there exist $\boldsymbol{\mu}^k \in \mathbb{R}_+^{A^k}$ and $\boldsymbol{\pi}^k \in \mathbb{R}^{V^k}$ for all k such that $(\mathbf{x}^k, \boldsymbol{\mu}^k, \boldsymbol{\pi}^k)_{k \in K}$ is a solution of the following augmented optimization problem (AMNEP(\mathbf{e})) with $\omega = 0$.

$$\begin{aligned}
 (\text{AMNEP}(\mathbf{e})) \quad & \min \quad \omega \\
 (2) \quad & s.t. \quad \begin{pmatrix} M & \mathbf{0} & \mathbf{0} \\ C & -I & \mathbf{e} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\mu} \\ \omega \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ M^T \end{pmatrix} \boldsymbol{\pi} = \begin{pmatrix} \mathbf{b} \\ -\boldsymbol{\beta} \end{pmatrix} \\
 (3) \quad & \mathbf{x} \cdot \boldsymbol{\mu} = 0 \\
 (4) \quad & \mathbf{x} \geq \mathbf{0}, \boldsymbol{\mu} \geq \mathbf{0}, \omega \geq 0, \boldsymbol{\pi} \in \mathbb{R}^{\sum_k |V^k \setminus \{s^k\}|},
 \end{aligned}$$

where M encodes the incidence matrices of the graphs (V^k, A^k) , C encodes the parameters $(\alpha_a^k)_{k \in K, a \in A^k}$ of the cost functions and $\mathbf{b} = (b_v^k)_{k \in K, v \in V^k}$ is the demand vector.

3.2. A Lemke-like algorithm. We define a *basis* for problem (AMNEP(\mathbf{e})) to be a subset B of indices such that the square matrix $\left(\begin{pmatrix} M & \mathbf{0} & \mathbf{0} \\ C & -I & \mathbf{e} \end{pmatrix}_B \middle| \begin{pmatrix} \mathbf{0} \\ M^T \end{pmatrix} \right)$ is nonsingular. Note that we extend the usual notion of basis used in linear programming, in order to deal directly with the unsigned variables ' π '.

Given a basis B we define the *basic solution* associated to B as the unique solution $(\bar{\mathbf{x}}, \bar{\boldsymbol{\mu}}, \bar{\omega}, \bar{\boldsymbol{\pi}})$ satisfying Equation (2) where each variable whose index is not in B is set to zero. Moreover, B is said to be *feasible*, resp. *complementary*, if the associated basic solution also satisfies Equation (4), resp. Equation (3) (under non-degeneracy assumption).

As for the simplex algorithm or the Lemke algorithm, the following lemma leads to a “pivoting” algorithm.

Lemma 3. *Given a feasible basis B and an index $i \notin B$, there is at most one feasible basis $B' \neq B$ in the set $B \cup \{i\}$. If there is no such basis, the polytope $\mathcal{P}(\mathbf{e})$ containing all feasible solutions of (AMNEP(\mathbf{e})) has an infinite ray.*

At each step, we have a current basis B^{curr} , we determine the entering index i , and we compute the new basis in $B^{curr} \cup \{i\}$, if it exists, which becomes the new current basis B^{curr} ; and so on. When we consider a complementary feasible basis, we can determine easily the entering index:

- (1) either the index of ω is not in the basis, and the problem is solved,
- (2) or there is exactly one pair (a_0, k_0) with $a_0 \in A^{k_0}$ such that neither the index of $x_{a_0}^{k_0}$ nor the index of $\mu_{a_0}^{k_0}$ are in the basis.

In this second case, when one of these two indices has just left the basis in the pivot algorithm, the other one is the entering index. If there is no infinite ray, called *secondary ray*, along the path, at the end the algorithm reaches a complementary feasible basis B^{end} where there is no such two indices. We are then in case (1), and have a solution of problem (AMNEP(\mathbf{e})) with $\omega = 0$, i.e. a solution of the Multiclass Network Equilibrium Problem.

It remains to find, with a good choice of \mathbf{e} , an initial feasible complementary basis and to show that there is no secondary ray.

Lemma 4. *There exists a vector \mathbf{e} such that*

- *We can compute a feasible complementary basis B^{ini} in polynomial time.*
- *There is a primary ray in $\mathcal{P}(\mathbf{e})$ originating at the feasible basic solution associated to B^{ini} .*
- *Under non-degeneracy assumption, there is no secondary ray in $\mathcal{P}(\mathbf{e})$.*

Combining the pivoting algorithm and Lemma 4 leads to the following result:

Theorem 2. *Under the non-degeneracy assumption, there is a Lemke-like algorithm solving the Multiclass Network Equilibrium Problem with affine costs.*

By perturbation arguments, this result provides a constructive proof of the existence of an equilibrium, even if the non-degeneracy assumption is not satisfied. It also shows that the problem of finding such an equilibrium belongs to the PPAD complexity class (see Papadimitriou [6]).

3.3. Computational experiments. The experiments are made on $n \times n$ grid graphs (Manhattan instances) where each pair of adjacent vertices is linked by an arc in each direction. The algorithm has been coded in C++ and tested on a PC Intel[®] Core[™] i5-2520M clocked at 2.5 GHz, with 4 GB RAM. Some computational results are given in the following table where each row contains average figures obtained on five instances on the same graph and with the same number of classes, but with various origins, destinations, and costs parameters.

Classes	Grid	Vertices	Arcs	Pivots	Algorithm (s.)	Total time (s.)
2	6×6	36	120	54	0.08	1.2
	8×8	64	224	129	0.9	8.9
4	4×4	16	48	41	0.06	0.66
	6×6	36	120	126	0.9	10.3
	8×8	64	224	249	5.4	55
10	4×4	16	48	107	0.7	8.9
	6×6	36	120	322	15	155
	8×8	64	224	638	87	857
50	2×2	4	8	56	0.3	5.5
	4×4	16	48	636	105	1127

The column “Algorithm” provides the time in seconds needed for the whole execution of the pivoting step, and the column “Pivots” contains the number of pivots performed. The column “Total time” gives the total time in seconds needed to solve the problem.

It seems that the number of pivots remains always reasonable. The essential computation time is spent on two matrix inversions, needed for the preparation of the pivoting step and for the final computation of the solution. The program has not been optimized. Since there are several efficient techniques known for inverting matrices or even methods for avoiding them, the results can be considered as very positive.

REFERENCES

- [1] M. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in Economics of Transportation*. Yale University Press, New Haven, CT, 1956.
- [2] H. Edelsbrunner. *Algorithm in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1987.
- [3] L. Gouvêas, J. Monnot, S. Moretti, and T. Nguyen. Congestion games with capacitated resources. In *SAGT*, 2012.
- [4] C.E. Lemke. Bimatrix equilibrium points and equilibrium programming. *Management Science*, 11:681–689, 1965.
- [5] I. Milchtaich. Congestion games with player-specific payoff functions. *Games Econom. Behavior*, 13:111–124, 1996.
- [6] C. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48:498–532, 1994.
- [7] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proc. Inst. Civil Engineers*, 2:325–378, 1952.

UNIVERSITÉ PARIS EST, CERMICS (ENPC), F-77455 MARNE-LA-VALLÉE
 E-mail address: thomas.pradeau@enpc.fr, frederic.meunier@enpc.fr